

and instructs generation engine 110 to process the selected report template 104 to generate report 118 (step 205).

In response, generation engine 110 loads the selected report template 104 and creates an environment for components 120 to operate (block 207). Generation engine 110 bi-directionally communicates with computing environment 106 and issues commands to computing environment 106 using the high-level programming language provided by computing environment 106. Computing environment 106 interprets and executes the commands as if they had been entered directly by user 103. In this manner, reporting components 120 can perform a variety of functions within the computing environment 106 including requesting data from the calculation workspace, evaluating an expression defined within the workspace, requesting data from simulator 122 and requesting data from a graphics package. In addition, reporting components 120 can change the state of the simulation model. For example, a reporting component can simulate the model, increment the model simulation by one or more time steps, change a parameter or initial condition of the model or even change the model itself by adding or removing a functional block. Reporting components 120 can also execute external processes that change the state of the calculation workspace.

While processing each component of the selected report template 104, generation engine 110 produces an intermediate representation 114 of report 118. This is advantageous because intermediate representation 114 is in a generic, abstract form that can be readily converted into a wide variety of document formats (block 207). In one embodiment intermediate representation 114 is formatted according to a generalized markup language such as SGML or XML.

Transformation engine 112 processes intermediate representation 114 to produce the report 118 in a user-selected document format such as Hypertext Markup web format (HTML), Microsoft Word interchange format (RTF), a typesetting language such as TeX, and Adobe Frame Maker interchange format (MIF) (block 209).

Figure 3 illustrates one embodiment of a graphical editor 300 displayed by user interface 102 by which user 103 can create and maintain report templates 104. Notably, editor 300 presents the hierarchy of components 305 defined by a corresponding report template. Using editor 300, user 103 can rearrange the order of the components 305, modify